



HPC Simulation of Magnetic Resonance Imaging

Tony Stöcker, Kaveh Vahedipour, N. Jon Shah

published in

Parallel Computing: Architectures, Algorithms and Applications,
C. Bischof, M. Bücker, P. Gibbon, G.R. Joubert, T. Lippert, B. Mohr,
F. Peters (Eds.),

John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **38**, ISBN 978-3-9810843-4-4, pp. 155-164, 2007.
Reprinted in: *Advances in Parallel Computing*, Volume **15**,
ISSN 0927-5452, ISBN 978-1-58603-796-3 (IOS Press), 2008.

© 2007 by John von Neumann Institute for Computing

Permission to make digital or hard copies of portions of this work for personal or classroom use is granted provided that the copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise requires prior specific permission by the publisher mentioned above.

<http://www.fz-juelich.de/nic-series/volume38>

HPC Simulation of Magnetic Resonance Imaging

Tony Stöcker, Kaveh Vahedipour, and N. Jon Shah

Institute of Medicine
Research Centre Jülich, 52425 Jülich, Germany
E-mail: {t.stoecker, k.vahedipour, n.j.shah}@fz-juelich.de

High performance computer (HPC) simulations provide helpful insights to the process of magnetic resonance image (MRI) generation, e.g. for general pulse sequence design and optimisation, artefact detection, validation of experimental results, hardware optimisation, MRI sample development and for education purposes. This manuscript presents the recently developed simulator JEMRIS (Jülich Environment for Magnetic Resonance Imaging Simulation). JEMRIS is developed in C++, the message passing is realised with the MPI library. The approach provides generally valid numerical solutions for the case of classical MR spin physics governed by the Bloch equations. The framework already serves as a tool in research projects, as will be shown for the important example of multidimensional, spatially-selective excitation.

1 Introduction

Bloch equation-based numerical simulation of MRI experiments is an essential tool for a variety of different research directions. In the field of pulse sequence optimisation, e.g. for artefact detection and elimination, simulations allow one to differentiate between effects related to physics and hardware imperfection. Further, if the simulation environment is able to simulate the hardware malfunction, then results may be used for the optimisation of the hardware itself. Another prominent application is the design of specialised RF pulses which is often based on numerical simulations of the excitation process. In general, the interpretation and validation of experimental results benefit from comparisons to simulated data, which is especially important in the context of MRI sample development, e.g. for the development of implants. Another important direction of application is image generation for the purpose of image processing – here, complete control of the properties of the input data allows a tailored design of image processing algorithms for certain applications.

The numerical simulation of an MRI experiment is, in its most general form, a demanding task. This is due to the fact that a huge spin ensemble needs to be simulated in order to obtain realistic results. As such, several published approaches have reduced the size of the problem in different ways. The most prominent method is to consider only cases in which analytical solutions to the problem exist¹. However, for the case of radiofrequency (RF) excitation in the presence of time varying gradient fields no analytical solution exists and, thus, the important field of selective excitation cannot be studied with such an approach. Apart from the computational demand, the complexity of the MRI imaging sequence is an additional obstacle. The difficulty of MRI sequence implementation using the software environments of commercial MRI scanner vendors – painfully experienced by many researchers and pulse programmers – can be significantly reduced with appropriate software design patterns.

The JEMRIS project was initiated taking all the aforementioned considerations into

account. It takes advantage of massive parallel programming on HPC (high performance computing) architectures. The aim of the project was to develop an MRI simulator which is general, flexible, and usable. In detail, it provides a general numerical solution of the Bloch equations on a huge ensemble of classical spins in order to realistically simulate the MRI measurement process. Further, it takes various important off-resonance effects into account.

2 Theory

2.1 MR Signal Evolution

The JEMRIS simulator is based on a classical description of MRI physics by means of the Bloch equations, describing the sample by its physical properties of equilibrium magnetisation, M_0 , and the longitudinal and transverse relaxation times, T_1 and T_2 , respectively. It provides an exact description for the magnetisation vector, $\mathbf{M}(\mathbf{r}, t)$, of non-interacting spin isochromates under the influence of an external magnetic field. For MRI, the field decomposes to the strong static component, B_0 , a temporally and spatially varying field along the same direction (the imaging gradients \mathbf{G}) and the orthogonal components of the RF excitation field, B_1 . The total field is thus given by

$$\mathbf{B}(\mathbf{r}, t) = [B_0 + \mathbf{G}(t) \cdot \mathbf{r}] \mathbf{e}_z + B_{1x}(\mathbf{r}, t) \mathbf{e}_x + B_{1y}(\mathbf{r}, t) \mathbf{e}_y. \quad (2.1)$$

A mathematical and numerical treatment is greatly simplified in the rotating frame of reference, in which the effect of the main field is not seen since the coordinate system rotates with the speed of the spin precession. Here, the formulation of the Bloch equation in cylindrical coordinates is very well suited for a numerical implementation

$$\begin{pmatrix} \dot{M}_r \\ \dot{\varphi} \\ \dot{M}_z \end{pmatrix} = \begin{pmatrix} \cos \varphi & \sin \varphi & 0 \\ -\frac{\sin \varphi}{M_r} & \frac{\cos \varphi}{M_r} & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \left[\begin{pmatrix} -\frac{1}{T_2} & \gamma B_z & -\gamma B_y \\ -\gamma B_z & -\frac{1}{T_2} & \gamma B_x \\ \gamma B_y & -\gamma B_x & -\frac{1}{T_1} \end{pmatrix} \cdot \begin{pmatrix} M_r \cos \varphi \\ M_r \sin \varphi \\ M_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{M_0}{T_1} \end{pmatrix} \right] \quad (2.2)$$

where γ is the gyromagnetic ratio of the nuclei (usually protons) under consideration. The complex MR signal is then described by the signal equation

$$S(t) \propto \int_V M_r(\mathbf{r}, t) \exp[i\varphi(\mathbf{r}, t)] d^3r \quad (2.3)$$

which integrates all components within the RF coil volume, V . For the description of the MR measurement process, the time evolution of each spin isochromate is completely different and the problem is ideally suited for numerical treatment with parallel processing. In MRI, the measurement process is expressed by the MRI sequence² which describes the timing of pulsed currents in various coils to produce the RF field for excitation and the gradient field for encoding spatial information to the phase/frequency of the MR signal: $\varphi(\mathbf{r}, t) = \gamma \int \mathbf{G}(t) dt \cdot \mathbf{r} \equiv \mathbf{k}(t) \cdot \mathbf{r}$. Inserting this expression in Eq. (2.3) shows that the MR signal, $S(t)$, can be reordered as a function of the k-vector, $S(\mathbf{k})$. Then, the MR image, $M_r(\mathbf{r})$, is given by the Fourier transformation of the acquired signal. Thus, the timing of the gradient pulses defines the so called k-space trajectory, $\mathbf{k}(t)$, along which the necessary information for image generation is acquired during the measurement. In

general the MR image, $M_r(\mathbf{r}) = M_r(M_0(\mathbf{r}), T_1(\mathbf{r}), T_2(\mathbf{r}))$, depends on the timing of the MRI sequence through the time evolution of the Bloch equation. In this way, images with various desirable properties such as soft-tissue contrast can be obtained. The image contrast is based on differences in the proton density, M_0 , and/or the relaxation times, T_1, T_2 , in biological tissue components; this unique feature provides a variety of medical imaging applications and it is the basis for the success of MRI in medicine.

2.2 Selective Excitation

Multidimensional, spatially-selective excitation⁴ is an important concept of growing interest in MRI, e.g. in the field of *in vivo* spectroscopy or for the challenging task of correcting subject-induced B_1 field inhomogeneities at ultra high fields⁵. However, thus far the computation of these pulses is based on a simplified physical model neglecting relaxation and off-resonance effects during the pulse. Under such conditions, the calculation of the unknown RF pulse shape, B_1 , from a given target excitation pattern, M_p , reduces to a linear system:

$$M_p(\mathbf{r}) = i\gamma M_0(\mathbf{r}) \int_0^T B_1(t) \exp[i\mathbf{r} \cdot \mathbf{k}(t)] dt \quad (2.4)$$

where M_0 is the equilibrium magnetisation and $\mathbf{k}(t)$ is a given k-space trajectory. A spatially and temporally discrete version of Eq. (2.4) can be solved for B_1 by suitable generalised matrix inversion methods. In contrast, the present approach provides a numerical method to design selective pulses under realistic experimental conditions. The simulator computes the effective transverse magnetisation, $M_e(\mathbf{r}, t)$, which is used to correct the RF pulse in order to account for effects not governed by Eq. (2.4). Thus, a minimisation problem is formulated and individually solved for all time steps $n\Delta t$ ($n = 1, \dots, N$), where $N\Delta t = T$ equals the pulse length:

$$\|M_p(\mathbf{r}) - M_e[M_0(\mathbf{r}), T_1(\mathbf{r}), T_2(\mathbf{r}), \Delta\omega(\mathbf{r}), B_1(t), \mathbf{k}(t)]\| = \min_{B_1} \quad (2.5)$$

Here, the difference between the desired magnetisation pattern and the effective magnetisation pattern is minimised with respect to the real and the imaginary parts of $B_1 = B_{1x} + iB_{1y}$. The starting point (B_{1x}, B_{1y}) for each of the N consecutive 2D minimisation problems is taken from the solution of Eq. (2.4). Note that the temporal sampling of B_1 is taken from the discrete version of Eq. (2.4), whereas the time evolution of the effective magnetisation is computed with much higher accuracy by the simulator, i.e. within each interval Δt the Bloch equation is individually solved for each spin to compute the norm in Eq. (2.5) for the minimisation routine. Once a minimum is found for the n -th step of the RF pulse, the final magnetisation states are taken as the starting condition for the next step.

3 Software Engineering

3.1 General Design of the Framework

The software design of JEMRIS had to meet two competing premises: Obviously, the object design had to reflect the physical world. At the same time the objects and members

were to remain highly maintainable, reusable and yet easy to handle.

A simple reflection of the class hierarchy is presented in Fig. 1. The object model in JEMRIS is based on the definition of the four main classes: *Sample*, *Signal*, *Model*, and *Sequence*, respectively. The *Sample* class describes the physical properties of the object, currently defined by the set $P = (M_0, T_1, T_2)$ at every spatial position $\mathbf{r} = (x, y, z)$ of the sample. Since MPI has no functionality for object serialisation, send/receive of sub-samples is realised with appropriate MPI data-types. Similarly, the *Signal* holds information about the ‘MR signal’ consisting of the net magnetisation vector $\mathbf{M}(t) = [M_x(t), M_y(t), M_z(t)]$ at every sampled time point. MPI functionality is implemented in the same way as for the *Sample* class. The *Model* class contains the functionality for solving the physical problem.

The design of the *Sequence* class and its underlying sequence framework proved to be a very demanding task. It supplies the most complex part of the simulator. For this a novel object-oriented design pattern for MRI is introduced which derives the basic parts, *Sequences* (loopable branch nodes) and *Pulses* (leaf nodes) from an abstract *Module* class. To further reduce the complexity in, and promote the encapsulation of, the objects in this part of the framework, an abstract factory approach using prototypes is implemented as suggested by Gamma et al⁶.

A sequence represents a left-right ordered tree. Fig. 2 depicts by example how the different modules of the well-known EPI sequence², loops and pulses, can be arranged in an ordered tree. Trees can be effectively accessed by recursion and are very well suited for building as well as accessing values of the sequence. The atomic sequences are containers for the pulses and, thus, display the functionality to emit pulses of various types. The pulses themselves are defined in a seceded class hierarchy.

The sequence tree is internally handled via XML. Thus, sequences themselves in turn can

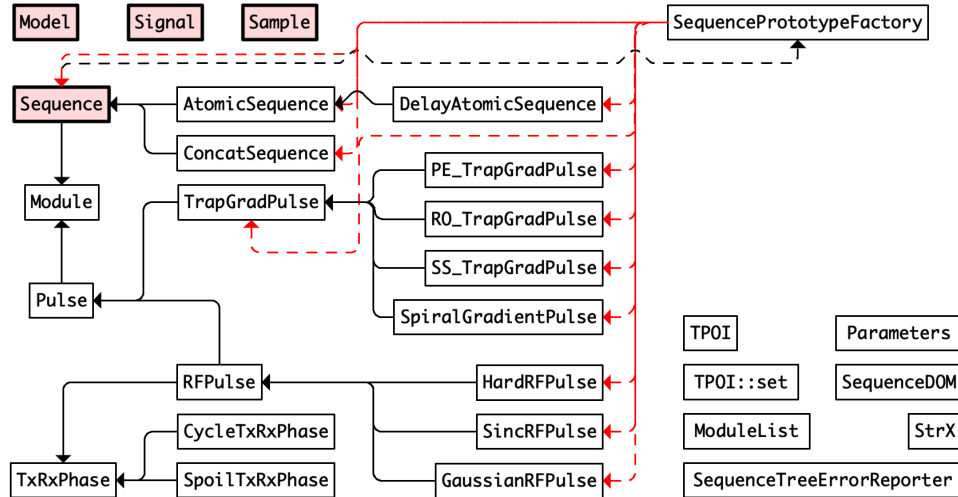


Figure 1. Class hierarchy of the basic JEMRIS components, *Sample*, *Signal*, *Model*, and *Sequence* class, respectively.

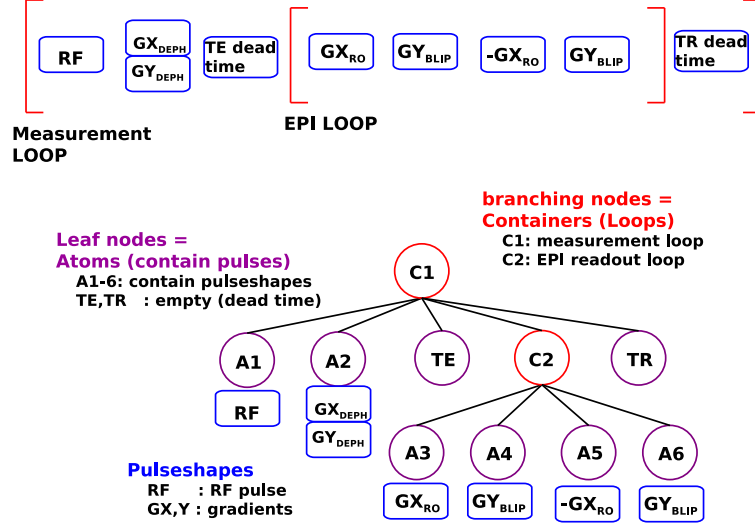


Figure 2. **Top:** Sketch of a native EPI pulse sequence diagram, consisting of an outer loop (e.g. slices) in which the RF excitation module, the dephasing gradients, dead times, and the inner loop for the EPI readout are nested. **Bottom:** Representation of the same diagram with a left-right ordered tree. Branching nodes represent loops in which other modules are concatenated. Leaf nodes represent the modules in which the actual pulse shapes are played out.

natively be read from, or written to, XML files. The Xerces C++ parser provided by the Apache Software Foundation^a is used for serialising and de-serialising the sequence object. Further, a GUI was implemented in MATLAB^b which allows one to interactively build as well as manipulate the sequence tree, view the pulse diagram, and perform the simulation. For the latter, MATLAB calls JEMRIS via ssh on the remote HPC site.

3.2 Parallel Implementation

Pseudo code of the parallel workflow of JEMRIS is shown in the parallel algorithm 1. (Comments are written with a reduced font size.) The (mostly sleeping) master process subdivides the problem in the beginning and harvests the result at the end. The basic functionality of the slave processes is hidden in the *Solve()* method of the *Model* class, where the solution for each spin is obtained during a recursive travel through the sequence tree. All functionality for introducing off-resonance effects is hidden in lines 9 and 19 of the slave process. Settings about this functionality, as well as the setting of the sample, is provided in the simulation XML file, which is parsed in the beginning.

The computation of selective excitation pulses utilising the simulator is shown in the parallel algorithm 2. The input parameters for the discrete problem consisting of N_t steps and

^a<http://xml.apache.org/xerces-c/>

^b<http://www.mathworks.com>

Algorithm 1 *Simulation Routine*

initialise N+1 MPI processes (n=0,...,N)

MASTER process (p=0)	SLAVE processes (p=1,...,N)
1: parse simulation XML file	1: parse simulation XML file
2: instantiate <i>Sample</i> object	2: instantiate <i>Model</i> object
3: split <i>Sample</i> into N <i>SubSample</i> objects	3: parse sequence XML file
4: for n=1 to N do	4: instantiate <i>Sequence</i> object
5: send n-th <i>SubSample</i> to n-th slave	→ 5: receive <i>SubSample</i>
6: end for	
	- functionality of <i>Model::Solve()</i> -
	6: instantiate <i>SubSignal</i>
	7: $N_s = \text{SubSample}::\text{getNumOfSpins}()$
	8: for $s = 1$ to N_s do
	9: instantiate <i>Spin</i> (s)
	10: <i>Sequence::Prepare()</i>
	11: $N_r = \text{Sequence}::\text{getNumOfRepetitions}()$
	12: $N_c = \text{Sequence}::\text{getNumOfChildren}()$
	13: for $r = 1$ to N_r do
	14: for $c = 1$ to N_c do
	15: if $N_c > 0$ (case <i>ConcatSequence</i>) then
	16: <i>Sequence</i> = <i>Sequence.getChild</i> (k)
	17: go to line 10: - recursion -
	18: else {compute solution}
	19: instantiate CVODE solver
	20: obtain <i>Solution</i>
	21: end if
	22: end for
	23: end for
	24: <i>SubSignal</i> += <i>Solution</i>
	25: end for
7: instantiate <i>Signal</i>	← 26: send <i>SubSignal</i>
8: for n=1 to N do	
9: receive <i>SubSignal</i> from n-th slave	
10: <i>Signal</i> += <i>SubSignal</i> (n)	
11: end for	
12: save <i>Signal</i>	

Algorithm 2 *Selective Excitation Routine*

initialise N+1 MPI processes (n=0,...,N)

MASTER process (p=0)	SLAVE processes (p=1,...,N)
1: parse selective excitation XML file	1: parse selective excitation XML file
	2: instantiate <i>Sequence</i> and <i>Model</i>
2: split and send <i>Sample</i> and <i>Target</i>	→ 3: receive <i>SubSample</i> and <i>SubTarget</i>
- loop over timesteps -	4: for $t=1$ to N_t do
3: for $t = 1$ to N_t do	5: bool $bCont = \text{true}$
4: bool $bCont = \text{true}$	
	- repeatedly call simulation routine -
- 2D conjugate gradient search for $B_1(t)$ -	6: while $bCont$ do
5: while $bCont$ do	7: <i>Model::Solve()</i> in time interval $[t - 1, t]$
6: for $n = 1$ to N do	- send difference between target and excited magnetisation -
7: receive ε_n from n-th slave	← 8: send $\varepsilon_n = \sum M_p(x_i) - M_e(x_i, t) $
8: end for	
9: $bCont = [\sum \varepsilon_n \neq \text{min}]$	
10: select next B_1 of the gradient search	
11: broadcast (B_{1x}, B_{1y}) and $bCont$	→ 9: receive (B_{1x}, B_{1y}) and $bCont$
12: end while	10: <i>Sequence.getChild</i> (t): <i>setB1</i> (B_{1x}, B_{1y})
	11: end while
13: store final $B_1(t)$	
	12: end for
14: end for	

the appendant solution of the linear problem in Eq. (2.4) is read from an XML file. After splitting the sample and the target pattern, the slave processes instantiate a sequence object which consists of a root node and N_t child nodes, each representing one time segment with a constant RF pulse (B_{1x}, B_{1y}) . Then, within a loop over the time steps, the master solves a conjugate gradient search to minimise the difference between the target pattern and the excited magnetisation by varying (B_{1x}, B_{1y}) . Thus, each slave repeatedly calculates its contribution to this difference each time with a new RF pulse at the current segment. Once a minimum is found, the master stores the result and continues with the next time step at which the slaves proceed with the final magnetisation state of the previous step.

4 Results

4.1 Benchmarks

The simulations were performed on a 16 dual-core CPU Opteron cluster. This can be seen as small-scale HPC; it is sufficient to perform 2D MRI simulations within minutes. Thus, the simulations presented here reduce to 2D examples, though the simulator is also able to treat 3D simulations. The performance of the simulator is depicted in Fig. 3. The results from sequential program profiling show, that most of the time is spent in the highly optimised CVODE libraries. However, a significant amount of time, 28 %, is spent in the *Sequence.getValue()* function, leaving room for future optimisation of data retrieval from the sequence tree. As expected, parallel performance scales nearly perfectly, i.e. the speed increases linearly with the number of processors.

4.2 MRI Simulation Examples

Fig. 4 shows an example of the GUI for sequence development, which allows interactive building of the sequence tree and various representations of the corresponding pulse diagram. The right part of Fig. 4 depicts the simulation GUI, showing an example of a simple

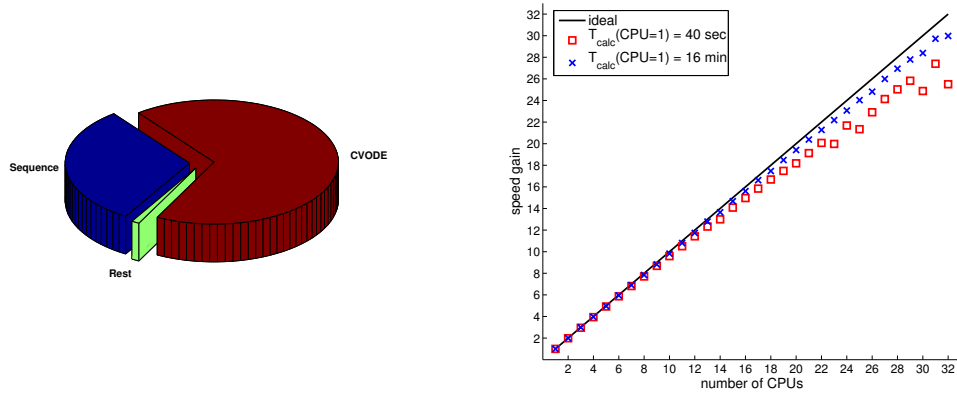


Figure 3. JEMRIS performance testing. **Left:** Sequential profiling shows that $\approx 70\%$ of the computing time is spent in the highly optimised CVODE library. **Right:** The speed gain due to parallelisation (right) is close to optimal for large scale problems, as expected.

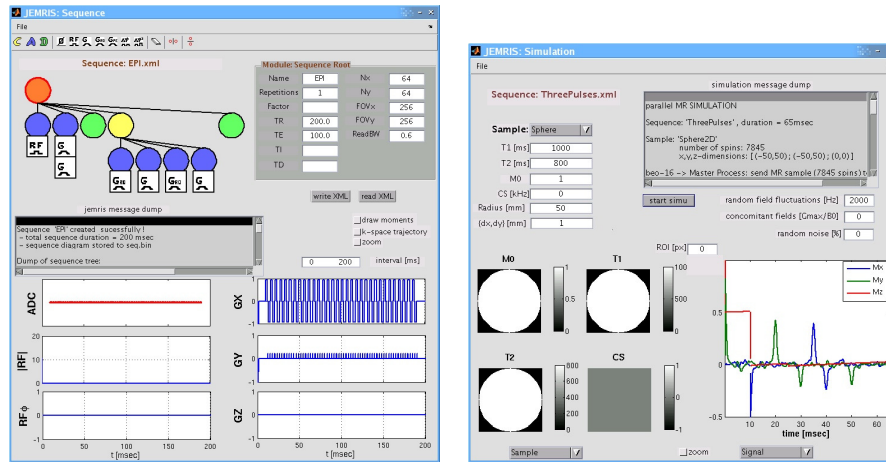


Figure 4. **Left:** Example of the JEMRIS GUI for sequence development showing the EPI sequence already shown in Fig. 2. In the top left, the sequence is interactively built and on the right the parameters of the individual nodes are set. In the bottom, the corresponding sequence diagram is shown. Possible error output is given in the slider-window. **Right:** Example of the JEMRIS GUI for simulations: a simple three pulse sequence (60° - 90° - 90° at 0-10-25 ms) applied to a homogeneous sphere under strong field inhomogeneities results in the well-established five echoes: three spin echoes, one reflected spin echo, and the stimulated echo.

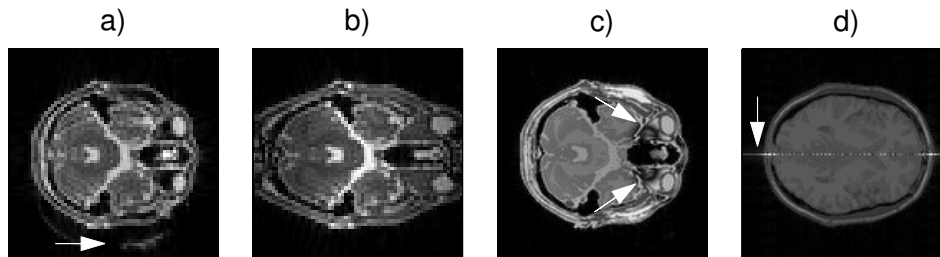


Figure 5. Example of artefact simulations on a human head model: **a** EPI with chemical shift. **b** EPI distortions due to a nonlinear read gradient. **c** TrueFISP banding artefacts resulting from susceptibility-induced field inhomogeneities. **d** Artefact in a spin echo sequence with a long refocusing pulse in the order of T_2 of the sample.

3-pulse experiment without any gradients involved. However, strong random field variations are applied yielding a very strong T_2^* effect, i.e. signal decays rapidly to zero. The well-established five MR echoes generated by such a three pulse experiment are visible. The tracking of magnetisation can be accurately performed at any desired time scale also during the application of RF pulses, since a generally valid numerical solution is computed. Examples of MRI artefact generation are given in Fig. 5: a) EPI simulation considering chemical shift effects yield the prominent fat-water shift in MRI; b) malfunctioning MR scanner hardware simulation with a nonlinear gradient field results in a distorted EPI image; c) TrueFISP simulation including susceptibility-induced field inhomogeneity yields the well-known banding artefacts in the human brain; d) artefact in spin echo imaging due

Figure	Number of Spins	Sequence	calculation time [min]
4 (right)	10.000	ThreePulses	0.1
5 a),b)	60.000	EPI	1
5 c)	60.000	TrueFISP	10
5 d)	60.000	Spin Echo	10

Table 1. Calculation times of the simulation examples. Note, that the calculation time strongly depends on the type of imaging sequence.

to a very long inversion pulse exciting transverse magnetisation. The corresponding calculation times are given in Table 1. Note that the last example, Fig. 5 d), cannot be realised with any simulator relying on analytical solutions of the Bloch equation due to neglect of relaxation effects during the RF pulse. The exact simulation of the simultaneous occurrence of RF pulses and imaging gradients with JEMRIS is the foundation for the derivation of new selective excitation pulses presented in the next section.

4.3 RF Pulse Design for Selective Excitation

For the demonstration of selective excitation, a homogeneous spherical object and a desired target magnetisation pattern were defined and the RF pulses of the common model and the new model were computed according to Eq. (2.4) and (2.5), respectively. Simulations were performed for $\approx 30,000$ spins. The results are depicted in Fig. 6. Note that the excitation patterns are not the result of a (simulated) imaging sequence but they are the effective

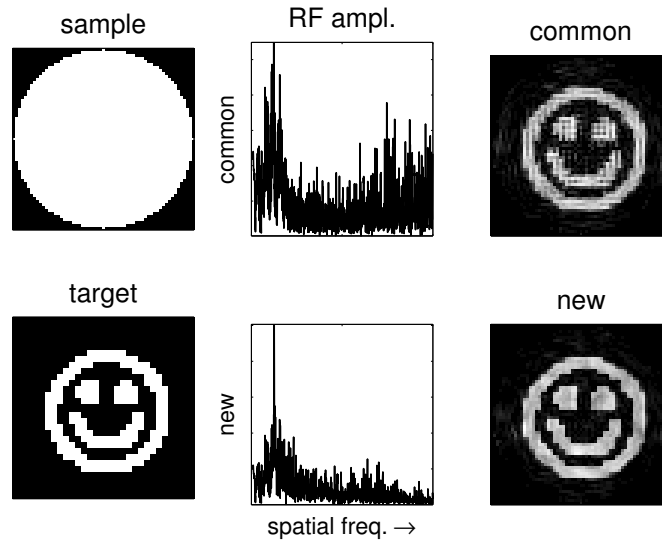


Figure 6. Example of multidimensional excitation. **Left Column:** 2D MR sample and the target pattern for selective excitation. **Middle Column:** RF pulses computed with the common approach and with the proposed method based on the JEMRIS framework. **Right Column:** Simulation results showing the excited spin system after applying the corresponding HF pulses.

patterns of excited spins directly after the pulse. In comparison, the optimised RF pulse computed with the new approach shows fewer higher spatial frequency components than the common approach. This is a well-known problem of the common linear approach, leading to high frequency artefacts on selectively excited images, as can be seen from the excited target pattern in Fig. 6. In comparison, the new approach excites a pattern that better approximates the target pattern.

5 Conclusion

JEMRIS performs (classically) general MRI simulations. It allows time tracking of the net magnetisation, separation of different effects, and therefore accurate investigation of MR samples (e.g. implants) and MRI sequences under a variety of experimental conditions. The parallel HPC implementation allows for the treatment of huge spin ensembles resulting in realistic MR signals. On small HPC clusters, 2D simulations can be obtained on the order of minutes, whereas for 3D simulations it is recommended to use higher scaling HPC systems. As shown, the simulator can be applied to MRI research for the development of new pulse sequences. The impact of these newly designed pulses in real experiments has to be tested in the near future. Further, the rapidly evolving field of medical image processing might benefit of a “gold standard”, serving as a testbed for the application of new methods and algorithms. Here, JEMRIS could provide a general framework for generating standardised and realistic MR images for many different situations and purposes. This aspect is especially interesting, since the image processing community usually has adequate access to HPC systems.

References

1. H. Benoit-Cattin, G. Collewet, B. Belaroussi, H. Saint-Jalmes and C. Odet, *The SIMRI project: a versatile and interactive MRI simulator*, Journal of Magnetic Resonance, **173**, 97–115, (2005).
2. E.M. Haacke, R.W. Brown, M.R. Thompson and R. Venkatesan, *Magnetic Resonance Imaging: Physical Principles and Sequence Design*, (Wiley & Sons, 1999).
3. S. Cohen and A. Hindmarsh, *CVODE, a stiff/nonstiff ODE solver in C*, Computers in Physics, **10**, 138–143, (1996).
4. J. Pauly, D. Nishimura and A. Macovski, *A k-space analysis of small tip-angle excitation pulses*, Journal of Magnetic Resonance, **81**, 43–56, (1989).
5. T. Ibrahim, R. Lee, A. Abduljalil, B. Baertlein and P. Robitaille, *Dielectric resonances and B_1 field inhomogeneity in UHFMRI: computational analysis and experimental findings*, Magnetic Resonance Imaging, **19**, 219–226, (2001).
6. E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns*, (Addison Wesley, 1995).